IN THE CLAIMS

Please cancel claims 9, 15, 18, 27, 29, and 31 without prejudice.

Please amend claims 1, 11-14, 16, 17, 20-26 and 28 as follows:

1.   (Currently Amended) A computer-implemented method for processing rendering data containing vertices, comprising:

defining a vertex cache as a software cache located within a transform and lighting module;

determining that a first vertex of the rendering data has already been transformed but not lighted and storing the first vertex in the vertex cache such that the first vertex bypasses a transformation module of the transform and lighting module;

transforming vertices of the rendering data containing vertices that have not already been transformed from model space into clip space; and

storing each of the continuing to store vertices of the rendering data that have already been transformed but not lighted in a the vertex cache as needed to facilitate a single streamline branched architecture that avoids processing duplication of the vertices.

2.   (Original) The computer-implemented method of claim 1, further comprising examining each of the vertices before lighting to determine whether to cull.

3.   (Original) The computer-implemented method of claim 2, further comprising discarding any vertices that are culled.

4.   (Original) The computer-implemented method of claim 3, further comprising continuing processing of any vertices that are not culled.

5.   (Original) The computer-implemented method of claim 1, further comprising generating coordinates for the vertices by performing lighting and texture generation and transformation.

6.   (Original) The computer-implemented method of claim 5, further comprising performing view frustum clipping on the coordinates after the lighting and texture generation and transformation.

7.   (Original) The computer-implemented method of claim 6, wherein the coordinates are normalized homogenous coordinate system (NHCS) clip space coordinates.

8.   (Original) The computer-implemented method of claim 1, further comprising using Direct3D for mobile as a rendering standard.

9.   (Canceled)

10.   (Original) A computer-readable medium having computer-executable instructions for performing the computer-implemented method recited in claim 1.

11.   (Currently Amended) A process for transforming and lighting rendering data, comprising:

  inputting rendering data in model space containing vertices;

  transforming vertices in the rendering data from model space to clip space to generate transformed vertices;

  culling a transformed vertex of the transformed vertices prior to processing by a lighting module after determining ~~whether to cull at least some of the vertices prior to lighting the rendering data~~ that the transformed vertex is not needed; and

  lighting each of the transformed vertices using the lighting module, except for the culled transformed vertex, to compute color and generate transformed and lighted vertices from the rendering data.

12.   (Currently Amended) The process as set forth in claim 11, ~~wherein determining whether to cull~~ further ~~comprises~~ <u>comprising</u> examining each of the <u>transformed</u> vertices <u>to determine whether they have previously been lighted</u>.

13.   (Currently Amended) The process as set forth in claim ~~12~~ <u>11</u>, further comprising determining ~~whether a vertices forms~~ <u>that the culled transformed vertex forms</u> a back face of a triangle.

14.   (Currently Amended) The process as set forth in claim 13, further comprising discarding the ~~vertices if it does form the back face of a triangle~~ <u>culled transformed vertex.</u>

15.   (Canceled)

16.   (Currently Amended) The process as set forth in claim ~~12~~ <u>11</u>, further comprising determining ~~whether a vertices~~ <u>that the culled transformed vertex</u> is outside of one view frustum clip plane.

17.   (Currently Amended) The process as set forth in claim 16, further comprising discarding the ~~vertices if it is outside of one view frustum clip plane~~ <u>culled transformed vertex</u>.

18.   (Canceled)

19.   (Original) One or more computer-readable media having computer-readable instructions thereon which, when executed by one or more processors, cause the one or more processors to implement the process of claim 11.

20.   (Currently Amended) A <u>computer-implemented process contained on a</u> computer-readable medium ~~having computer-executable instructions~~ for rendering graphics on an embedded device, comprising:

inputting 3D data containing vertices in model space;

transforming the 3D data into clip space <u>in a fixed-point format in a normalized homogenous coordinate system (NHCS) to obtain NHCS fixed-point format vertices</u>;

examining each of the <u>NHCS fixed-point format</u> vertices before lighting to determine whether to cull the <u>NHCS fixed-point format</u> vertices;

storing the <u>NHCS fixed-point format</u> vertices as needed in a vertex cache to provide a single streamline branched architecture that avoids processing duplication of the <u>NHCS fixed-point format</u> vertices; and

performing view frustum clipping of the <u>NHCS fixed-point format</u> vertices to generate an output of 2D screen coordinates <u>to render the graphics represented by the rendering data on the embedded device</u>.

21.   (Currently Amended) The ~~computer-readable medium~~ <u>computer-implemented process</u> of claim 20, wherein examining each of the <u>NHCS fixed-point format</u> vertices before lighting further comprises determining whether any of the <u>NHCS fixed-point format</u> vertices form a back face of a triangle and, if so, culling those <u>NHCS fixed-point format</u> vertices.

22.   (Currently Amended) The ~~computer-readable medium~~ <u>computer-implemented process</u> of claim 21, further comprising discarding each of the culled <u>NHCS fixed-point format</u> vertices and ~~continue processing~~ <u>continuing to process NHCS fixed-point format</u> vertices that have not been culled.

23.   (Currently Amended) The ~~computer-readable medium~~ <u>computer-implemented process</u> of claim 20, wherein examining each of the <u>NHCS fixed-point format</u> vertices before lighting further comprises determining whether any of the <u>NHCS fixed-point format</u> vertices are outside of one view frustum clip plane and, if so, culling those <u>NHCS fixed-point format</u> vertices.

24.   (Currently Amended) The ~~computer-readable medium~~ <u>computer-implemented process</u> of claim 23, further comprising discarding each of the culled

NHCS fixed-point format vertices and ~~continue processing~~ continuing to process NHCS fixed-point format vertices that have not been culled.

25.   (Currently Amended) The ~~computer-readable medium~~ computer-implemented process of claim 20, wherein the vertex cache is contained in software and not in hardware.

26.   (Currently Amended) The ~~computer-readable medium~~ computer-implemented process of claim 20, wherein the view frustum clipping is performed after a lighting and texture generation and transformation of the NHCS fixed-point format vertices.

27.   (Canceled)

28.   (Currently Amended) A transform and lighting module for preparing rendering data for rendering on an embedded computing device, comprising:

a transformation module that transforms vertices in the rendering data into clip space to generate transformed vertices;

a vertex cache implemented as a software cache and located within the transform and lighting module that stores ~~vertices~~ a first vertex contained in the rendering data such that the first vertex has previously been transformed but has not previously been lighted such that the first vertex is not processed by the transformation module; ~~and~~

a lighting module that computes color for each of the transformed vertices; ~~and~~

a culling module positioned after the transformation module and before the lighting module that ~~examines each of~~ culled a second vertex from the transformed vertices prior to ~~determine whether to send vertices to~~ processing by the lighting module after determining that the second vertex was not needed such that the second vertex is not processed by the lighting module;

a texture generation and texture transformation module that computes texture coordinates and transforms the texture coordinates into a fixed-point format in a

normalized homogenous coordinate system (NHCS) to obtain NHCS fixed-point format vertices in clip space; and

a view frustum module positioned after the lighting module and after the texture generation and transformation module that performs view frustum clipping of the NHCS fixed-point format vertices in clip space to generate output data that can be rendered for display on a display device of the embedded computing device.

29.   (Canceled)

30.   (Original) The transformation and lighting module as set forth in claim 28, wherein the culling module performs at least one of: (a) back face culling; (b) view frustum culling.

31.   (Canceled)